상호 연관 데이터(correlated data)의 브로드캐스트를 위한 prefetching

최정필*, 신성욱*

Prefetching for Broadcasting Correlated Data

Jung Pil Choi, Sung Wook Shin

Abstract

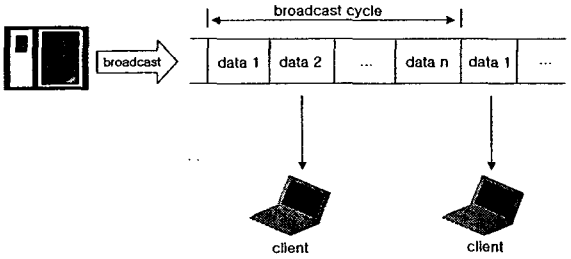
모바일 환경에서 브로드캐스트는 그 확장성 때문에 매우 유용한 데이터 전송 방법이다. "push-based" 데이터 전송 방식에서 서버는 넓은 대역폭을 통해 클라이언트에게 다양한 데이터를 반복적으로 브로드캐스트 한다.[1,2] 브로드캐스트에 기반을 둔 정보 시스템의 데이터 간의 연관성에 관한 연구는 미흡한 실정이다. 상호 연관 데이터의 브로드캐스트에서, 클라이언트는 자연스럽게 상호 연관된 데이터의 집합을 요청하게 되며, 데이터의 상호 연관성을 고려할 때 기존의 스케줄링 및 캐싱 기법 등은 달라져야 한다. CBS[3]에서는 모든 데이터 간의 연관도를 계산하여 최소 비용 경로를 구해, 이 순서대로 브로드캐스트하는 기법을 제안하였다. CBS 기법은, 클라이언트가 연관된 데이터를 동시에 요청하지 않고, NP-문제인 최소 비용 경로를 많은 데이터에 대해서 실시간에 계산해야 되며, 데이터 아이템 간의 상호 연관성이 클라이언트마다 다르게 정의되는 문제점이 있다. 따라서 본 논문에서는 응답 시간을 줄이기 위해, 브로드캐스트 되는 상호 연관 데이터의 prefetching 기법을 제안한다, 제안된 CT 기법은 상호 연관도와 브로드캐스트 대기 시간을 고려하여 캐시를 관리한다. CT를 현실적으로 적용한 ACT의 알고리즘을 소개하였으며, 시뮬레이션을 통해 CT의 성능과 특징을 실험하였다.

Key Words: data broadcast, data correlation, pretfech

* 고려대학교

1. 서론

모바일 컴퓨팅은 가장 빠르게 성장하고 있는 분야 중 하나이다. 이러한 모바일 컴퓨팅 분야의 주된 관심사는 사용자들에게 어떻게 효율적으로 데이터를 전달하는가의 문제이다. 공기를 전달 매체로 이용하는 모바일 환경에서, 브로드캐스트는 그 확장성 때문에 매우 유용한 데이터 전송 방법이다. "push-based" 데이터 전송 방식에서 서버는 넓은 대역폭을 통해 클라이언트에게 다양한 데이터를 반복적으로 브로드캐스트 한다.[1,2]



<그림 1> 브로드캐스트 기반 정보시스템

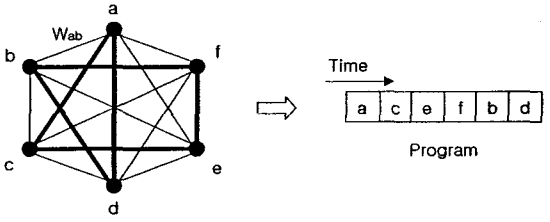
이러한 브로드캐스트에 기반을 둔 정보 시스템에 관한 연구가 많이 진행되었지만, 데이터 간의 연관성은 포함되지 않았다. 클라이언트는 자연스럽게 상호 연관된 데이터의 집합을 요청하게 되며, 데이터의 상호 연관성을 고려할 때 기존의 스케줄링 및 캐싱 기법 등은 달라져야 한다. CBS[3]에서는 모든 데이터 간의 연관도를 계산하여 최소 비용 경로를 구해, 비용의 순서대로 브로드캐스트하는 기법을 제안하였다. CBS 기법의 문제는 다음과 같다. 첫째, 클라이언트는 연관된 데이터를 동시에 요청하지 않는다. 둘째, NP-문제인 최소 비용 경로를 많은 데이터에 대해서 실시간에 구하는 것은 응답 시간을 증가시킨다. 셋째, 데이터 아이템 간의 상호 연관성은 클라이언트마다 다르게 정의되며, 따라서 이를 서버 측에서 지원해 주기 힘들다. 따라서 본 논문에서는 응답 시간을 줄이기 위해, 브로드캐스트 되는 상호 연관 데이터의 prefetching 기법을 제안한다.

2. 관련 연구

2.1 상호 연관 데이터 스케줄링

브로드캐스트 되는 데이터 사이에는 근본적으로 상호 연관이 존재하게 된다. 가령, 서버가 브로드캐스트하는 데이터 중에서 HTML 문서를 다운로드하는 상황에서, 클라이언트는 하나의 단일 데이터 아이템만을 다운로드 하는 것이 아니라, 일련의 상호 연관된 데이터를 요청한다. 이 때, 데이터 아이템 간의 상호 연관성이 클수록, 클라이언트가 이 데이터 아이템들을 함께 요청할 가능성도 커진다. CBS[3]에서는 이러한 상호 연관 데이터의 브로드캐스트 스케줄링 기법을 제안하였다.

CBS(Correlation-Based Scheduling) 전략에서 각 데이터 아이템 간의 상호 연관은 다음과 같은 네트워크 형태의 그래프로 표현할 수 있다.



<그림 2> CBS 예

CBS의 알고리즘은 다음과 같다.

1. 데이터 아이템을 나타내는 꼭지점과 간선으로 이루어진 전체 그래프는, 아이템 I와 j 사이의 간선이 $W_{ij}=1-P_{ij}$의 가중치를 갖도록 구성된다. P_{ij}는 클라이언트가 i와 j를 동시에 요청할 확률을 나타낸다.

$$(\sum_{j=i+1}^{M}\sum_{i=1}^{M} P_{ij}=1,\ P_{ij}=P_{ji})$$

2. 주어진 그래프에 대해서 traveling salesman problem(TSP)를 해결함으로써, 최소 비용을 내는 해밀턴 회로를 선택한다.
3. 선택된 회로를 따라, 모든 아이템을 브로드캐스트 프로그램에 넣는다.

그러나 이러한 CBS 기법은 몇 가지 문제점을 가지고 있다. 첫째, 클라이언트가 상호 연관 데이터를 동시에 요청하는 경우는 거의 없다고 볼 수 있다. 클라이언트가 동시에 요청하는 데이터 아이템은 독립된 개별 데이터 아이템이 상호 연관성을 갖는 것 보다는 항상 함께 요청되는 하나의 데이터 아이템으로 보는 것이 옳다. 따라서 상호 연관 데이터를 인접시켜 브로드캐스트하게 되면, 두 번째 데이터 아이템을 요청하는 시점에서, 요청된 아이템의 브로드캐스트 순서가 이미 지났으므로 이를 받기 위해서는 다음 브로드캐스트 주기까지 기다려야한다.

둘째, NP-문제인 최소 비용 경로를 많은 데이터에 대해서 실시간에 구하는 것은 응답시간을 증가시킨다. CBS에서 브로드캐스트 프로그램을 구성하는 알고리즘은 traveling salesman problem의 해법에 의존하며, 이는 잘 알려져 있는 NP-문제이다. 그림 2의 예는 6개의 데이터 아이템만을 가지는 단순한 예를 보이고 있지만, 실제 정보시스템에서는 상당한 수의 데이터 아이템을 대상으로 하게 된다. 이러한 환경에서 여러 개의 데이터 아이템에 대해 NP-문제를 푸는 것은 휴리스틱을 사용하더라도 상당한 계산량을 요구한다. 더군다나 데이터 아이템의 삽입과 삭제, 혹은 데이터 아이템 간의 상호 연관 정도의 변화가 빈번하게 일어나는 경우, 매번 브로드캐스트 프로그램을 다시 작성해야 하므로, 현실적으로 문제가 있다.

셋째, 데이터 아이템 간의 상호 연관성은 클라이언트마다 다르게 정의되며, 따라서 이를 서버 측에서 지원해 주기 힘들다. 일반적으로 데이터 아이템은 어느 정도 공통적인 액세스 확률과 상호 연관을 갖는다고 볼 수 있으나, 개별 클라이언트를 모두 만족시킬 수는 없다.

3. 상호 연관 데이터의 prefetching

3.1 ACT(approximation of CT)

PT 알고리즘[4]에서, 캐시에 들어 있는 각 데이터 아이템은 액세스 확률(P)와 브로드캐스트 프로그램에서 해당 데이터 아이템이 다시 나타날 때까지의 시간(T)의 곱인 PT 값을 갖는다. prefetching 시에는 캐시 내에서 제일 작은 PT 값을 갖는 데이터 아이템을 버린다. 그러나 O(캐시크기)의 수행 시간을 요구하는 최저 PT 값의 계산을, 매번 데이터 아이템이 브로드캐스트 될 때마다 수행하는 것은 현실

적이지 않다. 따라서 본 논문에서는 상호 연관 데이터의 prefetching을 위해서, APT (approximation of PT)[4]를 상호 연관 데이터에 대한 것으로 수정한 ACT(approximation of CT)를 사용하였다.

캐시에 들어 있는 데이터 아이템들은 context data(마지막으로 요청했던 데이터 아이템)과의 상호 연관도에 따라 그룹으로 묶이며, 각 그룹 내에 있는 데이터 아이템들은 같은 상호 연관도를 갖는 것으로 간주한다. 각 그룹은 하나의 큐(queue)를 구성하며, 하나의 큐 내부에서 요소들은 시간의 흐름에 따른 순서를 갖는다.

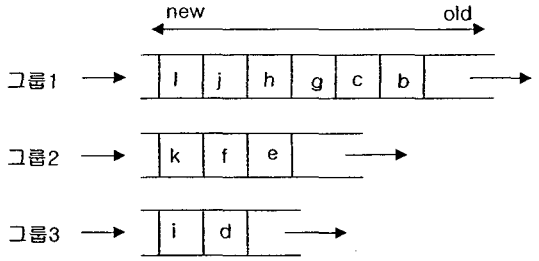
context data가 'a'라고 했을 때, 브로드캐스트가 진행됨에 따라 a와 연관된 데이터 아이템들을 prefetching하기 시작한다.

	a	b	c	d	e	f	g	h	i	j	k	l	m
a		7	8	1	4	6	9	9	2	7	5	8	5
b	7		5	4	3	4	9	8	2	4	1	8	9
c	8	5		5	8	1	2	6	5	5	7	4	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
m	5	9	1	4	5	8	4	5	6	3	2	1	1

<표 1> 데이터 아이템 간의 상호 연관도 예

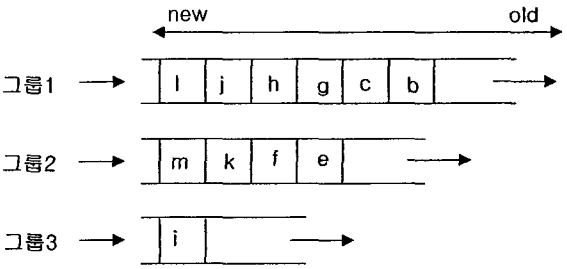
각 데이터 아이템 간의 상호 연관도가 <표 1>과 같을 때, 데이터 아이템 b~l을 순서대로 캐시에 넣으면 다음 그림과 같다.

예를 들어, $c_i \geq 7$인 데이터 아이템은 그룹1, $7 > c_i \geq 4$인 데이터 아이템은 그룹2, $4 > c_i$인 데이터 아이템은 그룹3으로 분류할 수 있고, 각 그룹1, 그룹2, 그룹3은 같은 상호 연관도를 갖는 것으로 추정한다. (c_i : 데이터 i와 context data와의 상호 연관도)



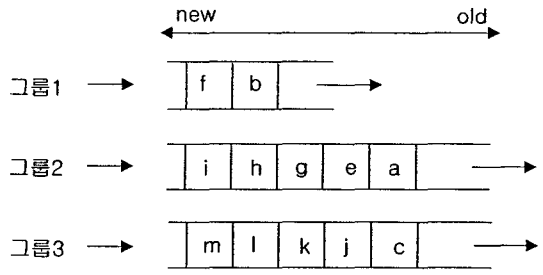
<그림 3> ACT의 그룹화 예

그리고 데이터 m을 prefetching하는 순간, 이를 캐시에 넣기 위해서는 캐시로부터 내보낼 데이터 아이템을 하나 선택해야 한다. 각 그룹의 큐에서, 큐 내의 데이터 아이템들은 모두 같은 연관도를 가지므로, 각 큐에서 가장 오래된 데이터 아이템인 b, e, d가 후보가 된다. 이 때, b, e, d의 CT값을 계산하여 가장 낮은 CT를 갖는 데이터 아이템을 버린다. m은 context data와 m이 갖는 상호 연관도에 따라 그룹이 배정된다. d를 버리고, $7 > c_m \geq 4$이므로 캐시는 다음 그림과 같이 된다.



<그림 4> m을 prefetching한 후의 캐시 구성

클라이언트에서 실제로 데이터 m이 요청되었을 때, m이 캐시 내에 존재하므로 이것을 그대로 사용하며, 캐시의 내용은 m에 대해 각 데이터 아이템이 갖는 상호 연관도에 따라 재구성된다. 이 때 context data도 캐시 내의 적절한 그룹으로 삽입된다.



<그림 5> m을 요청한 후의 캐시 구성

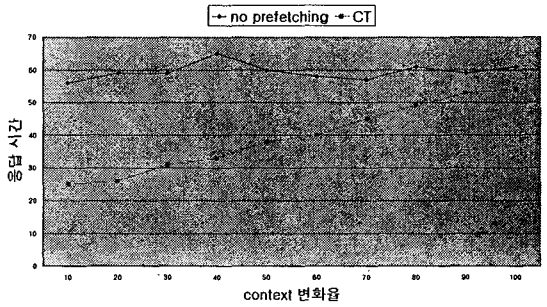
요청한 데이터 아이템이 캐시 내에 없는 경우, 해당 데이터 아이템이 브로드캐스트되기를 기다린다. 요청과 동시에 요청한 데이터 아이템에 대한 상호 연관성에 따라 캐시를 재구성하고, 이후부터는 요청한 데이터 아이템에 대해서 prefetching을 수행한다.

만약 요청한 데이터 아이템이 현재 캐시 내의 데이터 아이템과 상호 연관이 없는 경우에는, 같은 구조를 갖는 또 다른 일련의 그룹을 해당 데이터 아이템에 대해서 생성한다. 데이터 아이템의 삭제는 이전 그룹들에서 이루어지다가, 이전 그룹들에 있던 데이터 아이템이 모두 사라지게 되면 다시 원래의 prefetching 전략에 따라 데이터 아이템을 삭제한다.

3.2 시뮬레이션

120개의 데이터 아이템을 대상으로 하였으며, 각 데이터 아이템은 하나의 브로드캐스트 주기에서 한 번씩, 스케줄링을 고려하지 않고 임의의 순서로 브로드캐스트된다. 클라이언트는 전체 데이터 아이템의 10%에 해당하는 캐시를 가지며, 하나의 요청에 대한 응답을 한 후, 1~12사이의 임의의 단위 시간을 기다린 후 다음 요청을 수행하는 것으로 가정하였다.

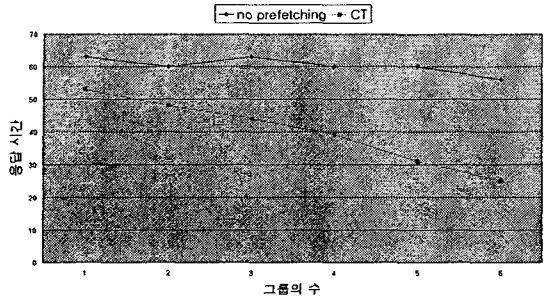
첫 번째 실험에서는 context의 변화에 따른 평균 응답 시간을 측정하였다. 전체 데이터 아이템은 6개의 그룹으로 나누어, 각 그룹 내에서만 상호 연관성을 갖는다. 상호 연관도는 그룹 내에서 1~10의 수치를 임의로 부여하였으며, 해당 그룹 외이 데이터 아이템과는 0의 상호 연관도를 갖는다. 데이터를 요청할 때는 context 변화율의 확률로 다른 그룹의 데이터 아이템을 요청하며, 나머지 경우, 그룹 내의 데이터 아이템을 요청한다.



<그림 6> context 변화율에 따른 응답 시간

먼저 prefetching을 수행하지 않는 경우는, context 변화율에 관계없이 60 전후로 나타났는데, 이는 전체 데이터 아이템의 절반에 해당한다. CT를 적용한 경우, 응답 시간이 전체적으로 단축되나, context 변화율이 커질수록 no prefetching에 가까워짐을 볼 수 있다. 이는 캐시의 크기가 작으므로, context가 임의로 변하는 경우 그 기능을 발휘하지 못하기 때문이다.

두 번째 실험에서는 context 변화율을 10%로 고정시키고, 그룹의 수를 1~6으로 변화시키면서 응답 시간을 비교해 보았다.



<그림 7> 그룹의 수에 따른 응답 시간

그룹의 수가 적을수록 응답 시간이 길어지고, 따라서 효율이 떨어지는 것을 볼 수 있다. 그룹의 수가 적으면 데이터 아이템 간의 상호 연관성이 상대적으로 축소되어, 그룹의 수가 1이 되면 상호 연관도를 고려하지 않는 경우와 같게 된다. 그래프에서 그룹의 수가 1일 때의 차이는 기본적으로 prefetching의 사용에 따른 것이다.

4. 결론

본 논문에서는 상호 연관 데이터가 갖는 의미를 재정의하고, 이에 따른 prefetching 기법을 제안하였다. CT prefetching 기법의 성능을 시뮬레이션을 통해 실험하였으며, 상호 연관 데이터의 브로드캐스팅은 context의 변화와 그룹을 결정하는 방법에 영향을 받음을 실험을 통하여 살펴보았다.

향후 연구로는 상호 연관 데이터에 대한 스케줄링 기법과 이에 따른 caching 및 prefetching 전략의 변화에 대한 연구가 필요하다.

참고문헌

[1] S. Acharya, R. Alonso, M. Franklin, S. Zdonik, "Broadcast Disks: Data Management for Asymmetric Communication Environments". Proceedings of the ACM SIGMOD Conference, 1995.

[2] S. Acharya, M. Franklin, S. Zdonik, "Dissemination-Based Data Delivery Using Broadcast Disks", IEEE Personal Communications, 1995.

[3] E. Yajima, T. Hara, M. Tsukamoto, S. Nishio, "Scheduling and Caching Strategies for Broadcasting Correlated Data". Proceedings of the ACM Symposium on Applied Computing (ACM SAC), 2001.

[4] S. Acharya, M. Franklin, S. Zdonik, "Prefetching from a Broadcast Disk", Proceedings of the ICDE, 1996.